CLAIMS

We claim:

1. A decoder circuit for decoding input data, said decoder implementing the maximum a posteriori probability decoding algorithm, comprising:

a table for computing the function $\log(e^{x_1} + e^{x_2})$ or $\ln(e^{x_1} + e^{x_2})$ where x_1 and x_2 are first and second argument values, each derived from said input data, said table storing a first data field including a plurality of table index values selected from a predefined range of $|x_1 - x_2|$ argument values, and a second data field including a plurality of computed table values computed based on the equation $\log(1 + e^{-|x_1 - x_2|})$ or $\ln(1 + e^{-|x_1 - x_2|})$ for each of said $|x_1 - x_2|$ argument values selected for said table index values;

an index value generation circuit for computing an index value $z = |x_1 - x_2|$ for addressing said table for a first argument value x_1 and a second argument value x_2, said index value generation circuit comprising:

a first inverter coupled to receive said second argument value x_2 for inverting the databits of said second argument value x_2;

an adder coupled to receive said first argument value x_1, said inverted second argument value x_2, and a value of one, said adder providing an output value equaling the difference between said first argument value x_1 and said second argument value x_2;

a second inverter coupled to receive said output value of said adder for inverting the data bits of said output value of said adder; and

a multiplexer having a first input terminal coupled to receive said output value of said adder, a second input terminal coupled to receive said inverted output value of said adder from said second inverter, and a control terminal coupled to receive a control bit;

wherein said multiplexer selects said first terminal as said index value z when said output value of said adder is a positive value and selects said second terminal as said index value z when said output value of said adder is a negative value.

2. The decoder circuit of claim 1, wherein said control bit is the most significant bit of said output value of said adder.

3. The decoder circuit of claim 1, wherein when said index value z is used to address said table, said index value z is compared with said plurality of table index values in said first data field to determine in which range said index value z corresponds.

4. The decoder circuit of claim 1, wherein said table is stored in a memory, and when said index value z is used to address said table, a predetermined number of address bits are extracted from said index value z for addressing said memory.

5. The decoder circuit of claim 1, wherein said plurality of table index values in said table are scaled by a first scaling factor, and said plurality of computed table values in said table are scaled by said first scaling factor.

6. The circuit of Claim 5, wherein said first argument value x_1 and a second argument value x_2 are not scaled by said first scaling factor.

7. The decoder circuit of Claim 1 wherein said circuit computes the function $\log(e^{x_1} + e^{x_2})$ or $\ln(e^{x_1} + e^{x_2})$ for said index value $z = |x_1 - x_2|$ by obtaining a first computed value from said plurality of computed table values in said second data field of said table associated with a first table index value of said plurality of table index values in said first data field corresponding to said index value z, and adding said first computed table value to the greater of said first argument value x_1 and said second argument value x_2.

8. A method in a decoder applying the maximum a-posteriori probability algorithm for computing the function $\log(e^{x_1} + e^{x_2})$ or $\ln(e^{x_1} + e^{x_2})$ for a first argument value x_1 and a second argument value x_2, comprising:

generating a table, comprising:

generating a first data field including a plurality of table index values being selected from a predefined range of $|x_1-x_2|$ argument values;

generating a second data field including a plurality of computed table values computed based on the equation $\log(1+e^{-|x_1-x_2|})$ or $\ln(1+e^{-|x_1-x_2|})$ for each of said $|x_1-x_2|$ argument values selected for said table index values;

computing an index value $z = |x_1 - x_2|$ for addressing said table for a first argument value x_1 and a second argument value x_2, comprising

inverting databits of said second argument value x_2,

adding said first argument value x_1, said inverted second argument value x_2, and a value of 1, thereby generating the difference of said first argument value x_1 and said second argument value x_2;

inverting databits of said difference of said first argument value x_1 and said second argument value x_2;

selecting said difference of said first argument value x_1 and said second argument value x_2 as said index value z when said difference is a positive value; and

selecting said inverted difference of said first argument value x_1 and said second argument value x_2 when said difference is a positive value.

9. The method of claim 8, wherein the most significant bit of said difference of said first argument value x_1 and said second argument value x_2 indicates whether said difference is a positive value or a negative value.

10. The method of claim 7, further comprising:

obtaining a first computed table value from said plurality of computed table values in said second data field of said table associated with a first table index value of said plurality of said table index values in said first data field corresponding to said index value z;

determining a greater of said first argument value x_1 and said second argument value x_2; and

adding said first computed table value to said greater of said first argument value x_1 and said second argument value x_2.

11. The method of claim 7, further comprising:

comparing said index value z to said plurality of table index values in said first data field of said table;

determining a first one of said plurality of table index values to which said index value z corresponds; and

obtaining a first computed table value from said plurality of computed table values in said second data field of said table associated with said first one of said plurality of table index values.

12. The method of claim 11, further comprising:

determining a greater of said first argument value x_1 and said second argument value x_2; and

adding said first computed table value to said greater of said first argument value x_1 and said second argument value x_2.

13. The method of claim 7, further comprising:

storing said table in a memory;

extracting address bits from said index value z for addressing said memory; and

obtaining a first computed table value from said plurality of computed table values in said second data field of said table associated with said address bits.

14. The method of claim 13, further comprising:

determining a greater of said first argument value x_1 and said second argument value x_2; and

adding said first computed table value to said greater of said first argument value x_1 and said second argument value x_2.

15. The method of claim 8, wherein said generating a table further comprises:

scaling said table index values by a first scaling factor; and

scaling said computed table values by said first scaling factor.